

Instrucciones:

- Cada pregunta múltiple opción contestada correctamente tiene un valor de 3,34 puntos. Cada pregunta incorrecta de la múltiple opción resta 1,11 puntos.
- **El puntaje total del examen es 100 puntos y se aprueba con 60 o más puntos.**

Múltiple Opción

1. En el área de ética profesional existen algunas cuestiones de responsabilidad profesional más allá del aspecto legal. Sobre estas cuestiones, indique la opción correcta:
 - a) Derechos de propiedad intelectual - implica conocer las leyes locales que rigen el uso de la propiedad intelectual, como patentes, derechos de autor, etc.
 - b) Confidencialidad - implica respetar la confidencialidad de los empleadores o clientes, aún cuando no haya suscrito un acuerdo de confidencialidad.
 - c) Competencia - implica no aceptar trabajos que están fuera del nivel de competencia,
 - d) **Todas las anteriores son correctas.**
2. Respecto a los casos de uso:
 - a) Una estrategia para identificarlos es: identificar primero a los actores, luego a los procesos del negocio que serán soportados por el sistema y por último definir casos de uso para las actividades en las cuales interactúan los actores y el sistema.
 - b) Si el sistema no tiene interacción con ninguna persona u otro sistema externo, entonces el propio sistema es el actor.
 - c) El sistema no es considerado como actor. Pues, un actor es una persona (o quizás otro sistema) que interactúa con el sistema para realizar un caso de uso.
 - d) **Dos opciones de las anteriores son correctas.**
3. Respecto a las técnicas de obtención de requisitos. Seleccione la opción correcta:
 - a) Las entrevistas son buenas para entender los requisitos de dominio.
 - b) **Las entrevistas son buenas para conseguir una comprensión global de qué quieren los stakeholders y cómo podrían interactuar con el sistema.**
 - c) No es necesario que los prototipos evolutivos sean construidos con calidad en el código y robustez desde el principio, si no que el objetivo es que se vayan mejorando en el transcurso del desarrollo.
 - d) Los Focus groups (grupo focal) son reuniones estructuradas en las cuales un selecto grupo de interesados y expertos trabajan en conjunto para definir, crear, refinar y acordar documentos y modelos que representen los requisitos de usuario.
4. Respecto a las historias de usuario. Seleccione la opción que NO es correcta:
 - a) Representan un requisito escrito de manera simple, sencilla y fácil de entender.
 - b) Los criterios de aceptación son detalles de la historia de usuario para que el equipo sepa lo que tienen que construir y lo que la contraparte espera.
 - c) Se recomienda que toda Historia de Usuario cumpla con 6 características que se pueden recordar bajo la regla mnemotécnica "INVEST".
 - d) **Si una historia de usuario está escrita con el formato: Como [rol] quiero [funcionalidad] para [beneficio] y es entendida por todos los integrantes del equipo de desarrollo, entonces no es necesario incluir criterios de aceptación para dicha historia de usuario.**
5. Respecto al modelado del sistema. Seleccione la opción NO correcta:
 - a) **Es el proceso de desarrollar modelos abstractos del sistema para ayudar a los analistas a entender sus funcionalidades, pero no facilita la comunicación con el cliente.**
 - b) Los modelos de sistemas en construcción son usados durante la ingeniería de requisitos para ayudar a explicar los requisitos propuestos a otros stakeholders del sistema.
 - c) Los modelos de contexto muestran lo que está fuera de los límites del sistema y un ejemplo de estos modelos son los diagramas casos de uso que incluyen la frontera del sistema.
 - d) Modelar la interacción entre los componentes ayuda a entender si la estructura propuesta del sistema es apropiada para los requisitos y atributos de calidad establecidos.

6. En relación a los requisitos indique la opción NO correcta:
- La forma de un requisito puede variar desde una oración con un alto nivel de abstracción hasta una especificación funcional matemática.
 - Los requisitos no son completos cuando existen conflictos o definiciones contradictorias entre ellos.**
 - Un único requisito no funcional, por ejemplo un requisito de seguridad, puede generar varios requisitos funcionales que definen servicios requeridos para el sistema.
 - Una regla de negocio corresponde a una política, guía, estándar o regulación que define o restringe algún aspecto del negocio.
7. En cuanto al diseño de software:
- Arquitectura en bajo nivel se refiere a la arquitectura de sistemas empresariales complejos que incluyen otros sistemas, etc.
 - (a) que se vincula con la arquitectura detallada en el diseño de programas individuales.
 - Arquitectura detallada tiene que ver con la arquitectura de los programas individuales.**
 - (c) y en este nivel nos preocupamos de que los programas individuales sean descompuestos en componentes.
8. Son ventajas de explicitar la arquitectura:
- Comunicación con los involucrados.
 - Análisis del sistema.
 - Reutilización a gran escala.
 - Todas las anteriores.**
9. Dadas las siguientes afirmaciones respecto al diseño de software indique la correcta:
- Dividir y conquistar, sugiere dividir el producto a construir en partes más pequeñas para facilitar el diseño.
 - Abstracción permite al diseñador considerar un componente sin preocuparse por los detalles de implementación (por ej. describiendo comportamiento exterior sin describir los detalles internos).
 - Modularidad es la característica de un sistema en el cual los componentes se pueden implementar separadamente procurando que el cambio en uno tenga impactos mínimos en el resto.
 - Todas las anteriores son correctas.**
10. Indique cuáles de las siguientes son características de los sistemas distribuidos:
- Compartir recursos - implica compartir programadores en más de un proyecto de desarrollo de software.
 - Apertura - implica que el código sea abierto para proveedores y clientes.
 - Concurrencia - implica procesamiento concurrente para mejorar rendimiento.
 - Escalabilidad - implica mayor rendimiento al agregar nuevos programadores.
 - Tolerancia a fallas - implica la capacidad de continuar en funcionamiento después de que se ha producido un error.
- Seleccione la opción correcta:
- III y V son correctas.**
 - Ninguna de las afirmaciones es correcta.
 - II, III y V son correctas.
 - Todas las afirmaciones son correctas.
11. Sobre construcción de software, indique la opción correcta:
- No depende del modelo de ciclo de vida utilizado.
 - Se relaciona con el orden en el cual se crean los componentes, se integran, se gestiona la calidad de la construcción y la asignación a los distintos programadores.**
 - (b) donde las pruebas de desempeño tienen como objetivo corregir defectos introducidos por los programadores.
 - Ninguna de las anteriores.
12. Un equipo está trabajando en un proyecto de duración de 20 semanas y costo USD 30.000 . En la semana 12, calculan el CPI y el valor obtenido es 0,8 (QUE VANE REVISE)
- El proyecto está siendo más costoso de lo planificado.
 - (a) y si sigo con la misma tendencia el proyecto me va a costar más de USD 37.000.**
 - El proyecto está siendo más lento de lo esperado.
 - (c) y si sigo con la misma tendencia el proyecto va a terminar en la semana 25

13. Con respecto a la estrategia de evolución de un sistema heredado:
- El valor de un sistema heredado para el negocio es un factor importante en la elección de la estrategia de su evolución; para evaluar cuál es el valor del sistema para el negocio hay que basarse principalmente en la opinión de los gerentes de la organización, ya que son los que tienen la visión general del negocio en su repartición.
 - Si el sistema heredado es de baja calidad siempre debe ser desechado completamente.
 - Si el sistema heredado es de alta calidad y además tiene alto valor para el negocio, se recomienda continuar manteniendo el sistema.
 - La cantidad de usuarios que utilizan un sistema heredado y la frecuencia con que lo usan son dos factores que inciden directamente en el valor del sistema para el negocio.
 - Para evaluar el valor de un sistema heredado para el negocio debe tenerse en cuenta, entre otros factores, la confianza de los usuarios en el sistema.

Seleccione la opción correcta:

- Sólo (I), (II) y (III) son correctas.
 - Sólo (I), (III) y (V) son correctas.
 - Sólo (III), (IV) y (V) son correctas.**
 - Sólo (I), (II), (III) y (IV) son correctas.
14. Cuando hablamos de evolución del software podemos identificar que un sistema se encuentra en distintas fases de su proceso de vida útil.
- Una de estas fases o estados es cuando el sistema está en Servicio. El sistema se encuentra en su uso operacional y los únicos cambios que se admiten son con el fin de que permita que el software continúe operativo. No se agregan nuevas funcionalidades.**
 - Una de estas fases o estados es cuando el sistema está en Servicio. El sistema se encuentra en su uso operacional y evoluciona únicamente cuando hay nuevos requisitos.
 - Una de estas fases o estados es cuando el sistema está en Retiro gradual. El sistema se encuentra en su uso operacional y evoluciona únicamente cuando hay nuevos requisitos.
 - Una de estas fases o estados es cuando el sistema está en Evolución. El sistema se encuentra en su uso operacional y evoluciona únicamente cuando hay que corregir defectos identificados en producción.
15. La gestión de sistemas heredados (o legados) debe considerar los siguientes aspectos:
- Cuando el sistema tiene alta calidad pero un bajo valor para la empresa siempre debe ser desechado.
 - Se debe evaluar el valor que tiene el sistema para el negocio. Para esto, se deben tomar en cuenta las opiniones de usuarios finales y proveedores de productos sustitutos.
 - Cuando el sistema tiene baja calidad pero un alto valor para la empresa, se trata de un sistema muy importante pero caro de mantener. Por esto, es recomendable que sea reestructurado o remplazado por alguna opción comercial disponible.**
 - Debe evaluarse la calidad del sistema. Esto se logra calculando la cantidad de errores que el sistema tendrá en el próximo año y entrevistando de forma diaria a la alta gerencia.
16. Los cambios urgentes:
- Deben siempre pasar por el comité de control de cambios y deben ser priorizados.
 - Surgen a partir de una falla grave y a partir de nuevas funcionalidades que no fueron previstas al especificar los requisitos originales.
 - También son conocidos como refactorización y son realizados de forma automatizada mediante integración continua.
 - Es posible que deban ser implementados sin pasar por todas las etapas del proceso de gestión de cambios. En general, la solución implementada no es la mejor. Por lo tanto, una estrategia para abordar esto, es registrarlo como un cambio de tipo no urgente para realizar más adelante.**
17. Sobre la gestión de riesgos, seleccione la opción correcta:
- Si uno de los riesgos de mi proyecto es un módulo que debo integrar a mi sistema, si no utilizo ese módulo, estoy aplicando la estrategia evitar.
 - a) y mantener reuniones periódicas con el cliente para ir mostrándole los avances y obteniendo retroalimentación es mitigar.
 - b) y subcontratar parte del trabajo a realizar a una compañía experimentada es transferir.**
 - c) y asignar a la persona con mayor experiencia a trabajar en el componente más crítico del sistema es aceptar.

18. En un proceso de desarrollo de software:
- La verificación es imprescindible, no así la validación.
 - La validación es imprescindible, no así la verificación.
 - La verificación y validación evalúan los mismos aspectos del software, solo que desde diferentes puntos de vista.
 - Ninguna de las anteriores es correcta.
19. Respecto a las etapas del desarrollo de un equipo. Seleccione la opción correcta:
- Todos los equipos, tarde o temprano, alcanzan la etapa de desempeño.
 - En la etapa de normalización, los miembros del equipo comienzan a trabajar conjuntamente y a ajustar sus hábitos y comportamientos para apoyar al equipo.
 - Cuando un equipo llega a un etapa, por ejemplo normalización, no ocurre que vuelva a una etapa anterior. Por ejemplo, turbulencia.
 - En la etapa de normalización los equipos funcionan como una unidad bien organizada.
20. Seleccione la opción correcta:
- Un proyecto es un esfuerzo temporal que se realiza para crear un servicio o resultado único.
 - a) y un proyecto solo finaliza cuando se entrega el producto al cliente.
 - a) y la EDT la realiza el director del proyecto y luego se la presenta al equipo para que programen.
 - a) La EDT debe reflejar dependencia entre los componentes, por ejemplo cual debe realizarse primero.
21. Dada la siguiente tablas de dependencias de actividades del proyecto, si se me solicita aplicar la técnica de crashing, sobre qué tarea la debería aplicar. Seleccione la opción correcta:

Actividad	Duración	Actividades de las que depende
A	2	-
B	5	A
C	4	A
D	10	B
E	8	B
F	6	E
G	4	D,F,H
H	18	A

Seleccione la opción correcta:

- D
 - E
 - C
 - H
22. En relación a Scrum:
- Es un método ágil que hace especial énfasis en el uso de las prácticas ágiles de programación por pares, refactoring y el desarrollo dirigido por pruebas.
 - En la fase inicial de scrum se genera un esquema de la planificación en donde se establecen los objetivos generales del proyecto y el diseño de la arquitectura del software en alto nivel.
 - El equipo de desarrollo de scrum es auto-gestionado. Por tal motivo es el propio equipo quien se encarga de gestionar el *backlog* de trabajo a ser realizado y no el cliente.
 - A diario se realiza una reunión corta entre todo el equipo (*daily meeting*). Esto permite que todo el equipo conozca que está pasando.
 - Un "sprint" es una iteración de desarrollo de scrum. Los sprints por lo general tienen una duración de 1 a 4 semanas.

Seleccione la opción correcta:

- Sólo (II), (III) y (V) son correctas.
- Sólo (I), (III), (IV) y (V) son correctas.
- Sólo (II) y (IV) son correctas.
- Sólo (II), (IV) y (V) son correctas.

23. En relación a la mejora de procesos:
- Tiene como objetivo mejorar la calidad del software , reducir los costos y/o el tiempo de desarrollo.
 - El enfoque ágil para la mejora se enfoca en mejorar los procesos y la gestión de los proyectos mediante la introducción de prácticas ágiles de ingeniería de software.
 - (a) y para poder evaluar la mejora del proceso se necesita recolectar datos.**
 - (b) y el ciclo de mejora de procesos consiste en: medir, analizar y cambiar.
24. Podemos identificar distintos niveles de pruebas. Acerca de las estrategias asociadas puede decirse que:
- En el nivel de pruebas de componentes (o unitarias) las técnicas estáticas permiten experimentar que el comportamiento del componente es el esperado.
 - (a) y una técnica dinámica es la técnica de caja negra, la cual está basada en la especificación del programa.
 - El segundo nivel de pruebas comprende las pruebas de integración donde para elegir una estrategia se debe considerar optimizar el tiempo requerido y el costo del entorno de pruebas.
 - (c) y una estrategia incremental es bottom-up, en la que deben utilizarse drivers para poder realizar las pruebas.**
25. Acerca de las inspecciones de artefactos de software se conoce que:
- Es la técnica menos formal y, en general, es realizada ad-hoc por el propio autor del artefacto a inspeccionar.
 - El artefacto es recorrido de forma informal en busca de defectos. Se lleva a cabo en una instancia donde se invita a todos los interesados técnicos y se aprovecha a educar a la audiencia sobre el producto.
 - Son realizadas por una herramienta automatizada que busca defectos o partes propensas a contener defectos.
 - Es un proceso formal donde se examina el artefacto en busca de defectos y elementos que puedan resultar en un mal comportamiento o no sean claros. Se suele utilizar una lista de verificación (checklist) y es posible realizar una inspección sobre un mismo artefacto desde distintos roles o perspectivas (seguridad, diseño, etc.).**
26. En relación a la técnica de derivación de casos de prueba a partir de casos de uso:
- Los flujos principales y alternativos se representan como diferentes caminos en el grafo de flujo de control del CU.
 - En caso de que un flujo alternativo pueda ejecutarse N veces (se visualiza con un bucle en el grafo de flujo de control), es suficiente con generar un caso de prueba para cubrir dicho flujo.
 - En caso de que un escenario pueda ejecutarse a través de diferentes combinaciones de condiciones (varios escenarios-condición), es recomendable generar al menos un caso de prueba para cada combinación de condiciones.
 - Al momento de generar los casos de prueba se recomienda generar la menor cantidad de casos de prueba que comprendan la mayor cantidad de condiciones inválidas posibles.
 - Al momento de generar los casos de prueba se recomienda generar la menor cantidad de casos de prueba que comprendan la mayor cantidad de condiciones válidas posibles.
 - Las condiciones que hacen posible la ejecución de un escenario pueden ser: datos de entrada, datos de contexto de la aplicación, datos que se ingresan durante la ejecución del caso de prueba, entre otros.
- Seleccione la opción correcta:
- Sólo (I), (II), (III) y (IV) son correctas.
 - Sólo (II), (III), (V) y (VI) son correctas.
 - Sólo (I), (III), (V) y (VI) son correctas.**
 - Todas las afirmaciones son correctas.
27. Dado un escenario de un proceso de desarrollo de software con restricciones de tiempo y recursos, frente a una liberación de una nueva versión de un producto existente, la cual incluye la corrección de bugs y funcionalidades nuevas, al momento de verificar esa nueva versión es importante tener en cuenta los siguientes aspectos para la planificación:
- Priorización de las funcionalidades (features) / correcciones(fixes) a verificar de acuerdo al riesgo estimado (de que falle). Las más riesgosas deben probarse primero.
 - Recursos disponibles (hardware/software, recursos humanos: habilidades, experiencia).
 - Ciclo de vida del software en el cual está inmersa esta liberación.
 - Tipo de características/funcionalidades a probar y tipo de producto a verificar.
 - Nivel de profundidad requerido para las pruebas (poca profundidad: solamente "casos felices", mucha profundidad: casos felices, casos inválidos, robustez, performance).

Seleccione la opción correcta:

- Sólo (I), (II), y (V) son correctas.
- Sólo (I), (III) y (IV) son correctas.
- Sólo (II), (III), (IV) y (V) son correctas.
- Todas son correctas.**

28. En relación a los métodos ágiles:
- I. Fueron originalmente desarrollados para la construcción de sistemas de pequeño y mediano porte.
 - II. Los métodos ágiles buscan, entre otras cosas, evitar generar documentación que a la larga seguramente no se use.
 - III. Han demostrado ser exitosos para el desarrollo de sistemas con varios *stakeholders* externos y con equipos distribuidos geográficamente.
 - IV. Al utilizar la técnica de programación por pares, siempre que sea posible se debe intentar trabajar con el mismo par.
 - V. En los procesos ágiles, al igual que en cualquier otro proceso de desarrollo de software, se deben realizar actividades de gestión, para asegurarse que se está haciendo el mejor uso del tiempo y recursos disponibles del equipo.

Seleccione la opción correcta:

- a) Sólo (I), (II), (III) y (IV) son correctas.
 - b) Sólo (I) y (V) son correctas.
 - c) **Sólo (I), (II) y (V) son correctas.**
 - d) Sólo (II), (III), (IV) y (V) son correctas.
29. Respecto de las técnicas de verificación basadas en la experiencia:
- a) Es un enfoque sistemático que se basa fuertemente en la intuición, habilidades y conocimiento del tester.
 - b) **Se basa en datos históricos o en *error-guessing*.**
 - c) Requiere de una documentación base para poder comenzar a verificar.
 - d) Los errores o experiencia del pasado no son válidos si el software cambia mucho entre una iteración y otra.
30. Seleccione la opción correcta:
- a) La reutilización de componentes rara vez se utiliza en los procesos de desarrollo de software.
 - b) Los productos genéricos (COT's) tienen la ventaja de que no necesitan mantenimiento, solamente basta con configurarlos adecuadamente.
 - c) Cuando se utiliza una estrategia de integración y configuración de COT's esta no requiere que los requisitos sean flexibles o que puedan ser re-negociados ya que los altos grados de configuración permiten satisfacer adecuadamente las necesidades del cliente.
 - d) **En general, para un mismo conjunto de requisitos, un software desarrollado a medida tiene un costo de parametrización y configuración menor en comparación a la adquisición de un COT.**